Fig 1

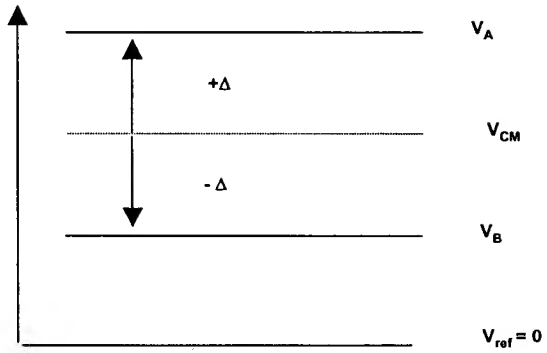


Fig 2

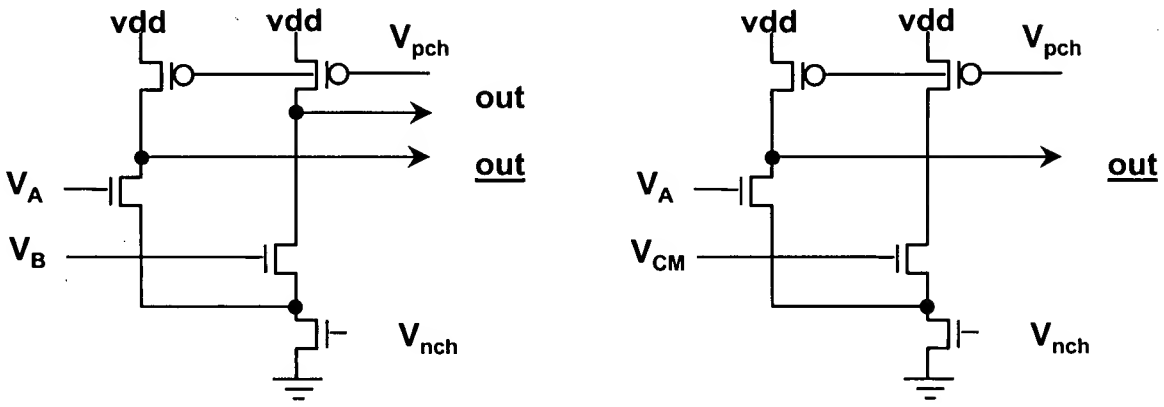
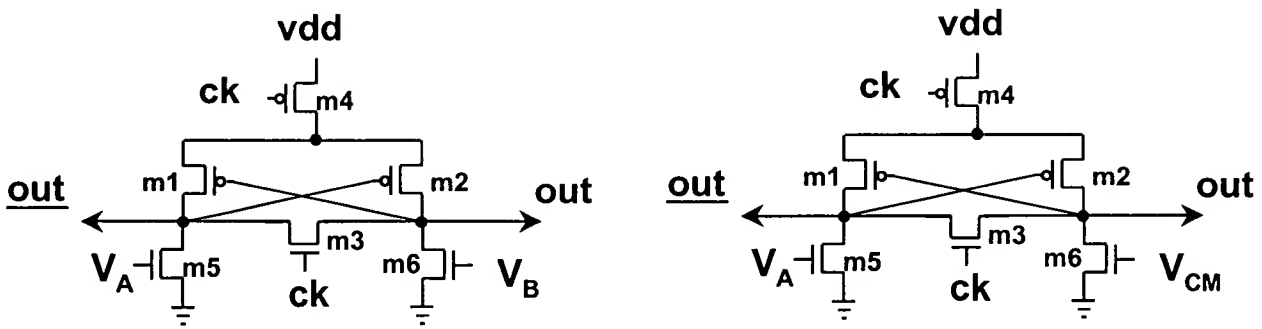
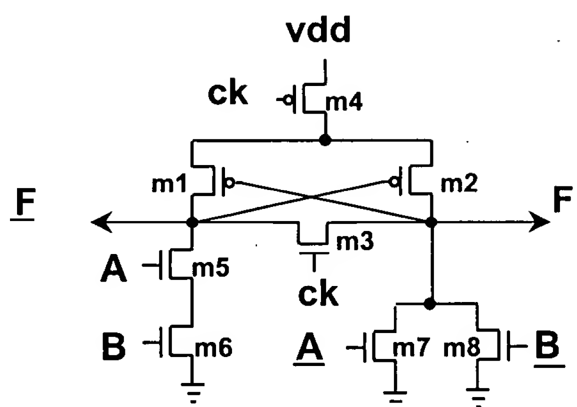


Fig 3



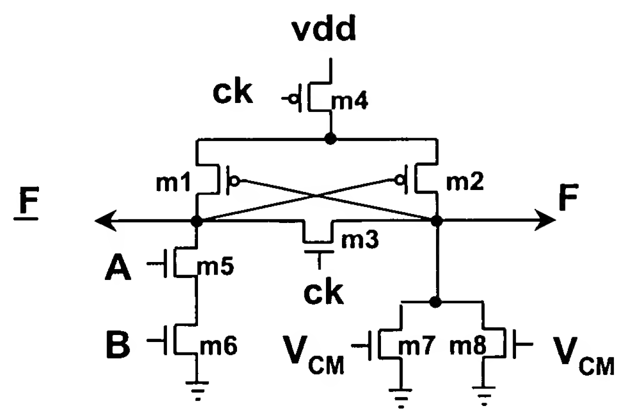
(a) Prior Art (b)

Fig 4



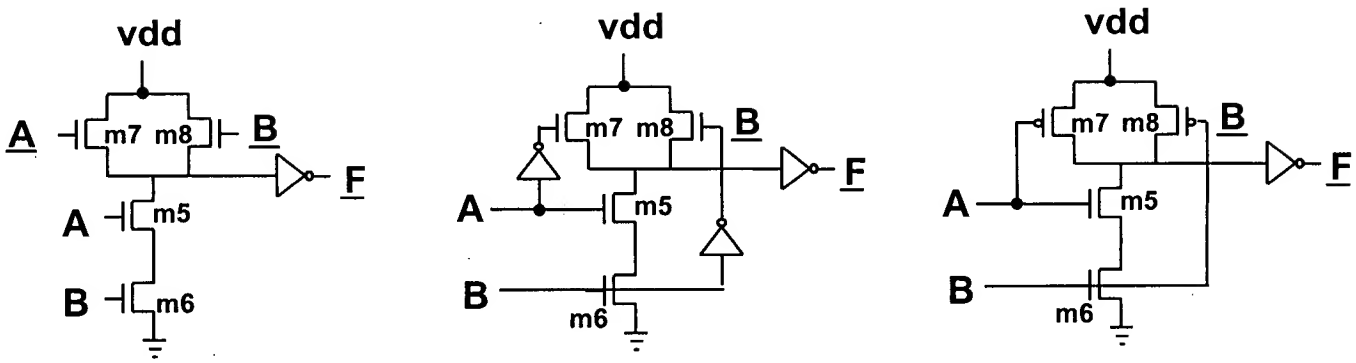
(a)

Prior Art



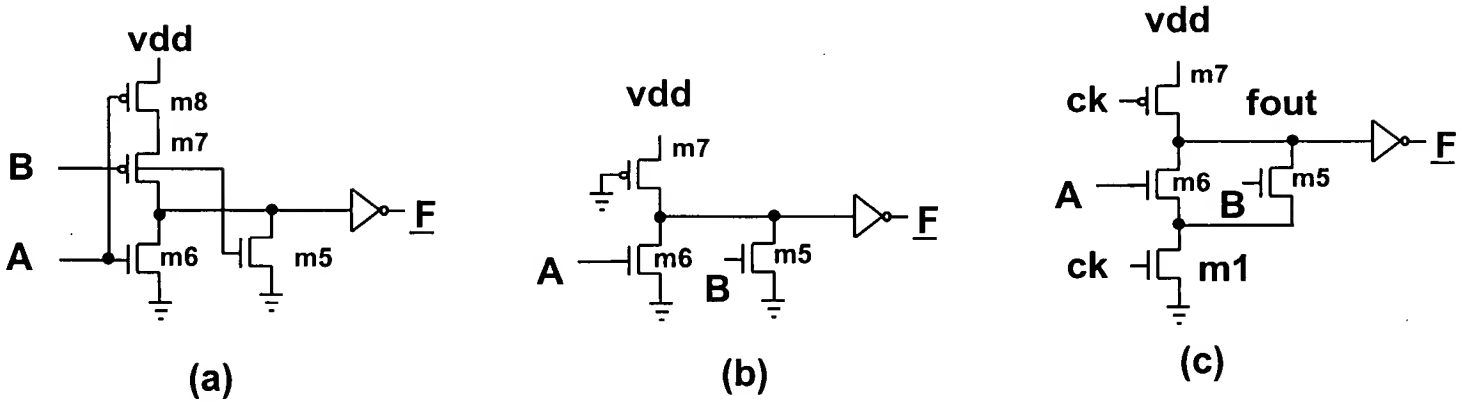
(b)

Fig 5



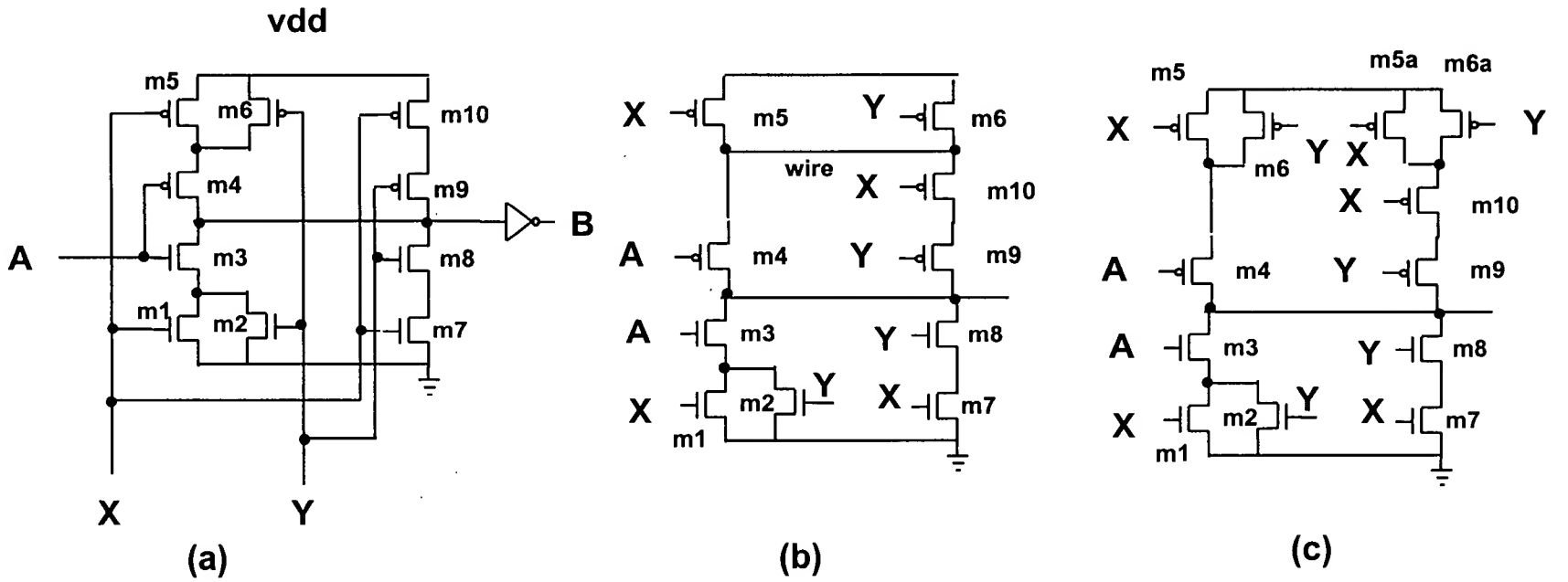
Prior Art

Fig 6



Prior Art

Fig 7



Prior Art

Fig 8

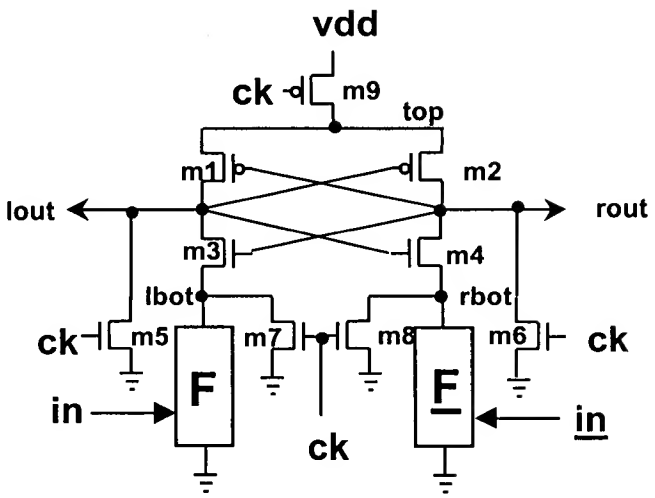

vdd
ck
m9
top
m1
m2
lout
rout
m3
m4
lbot
rbot
ck
m5
m7
m8
m6
ck
F
F
in
ck
in

Fig 9

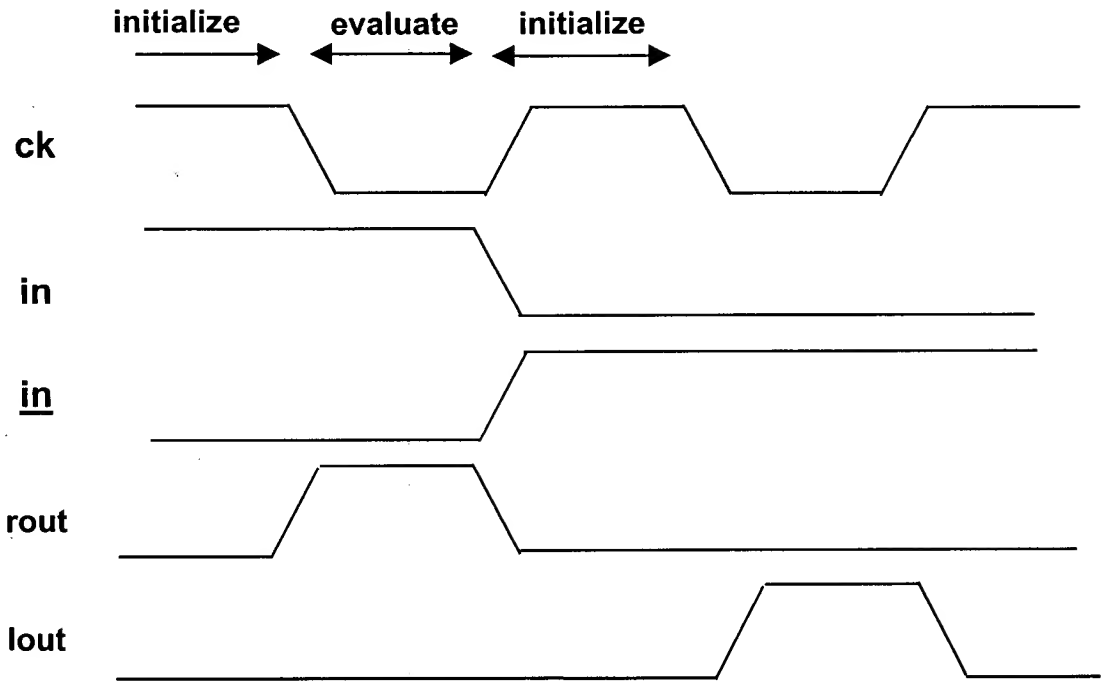


Fig 10A

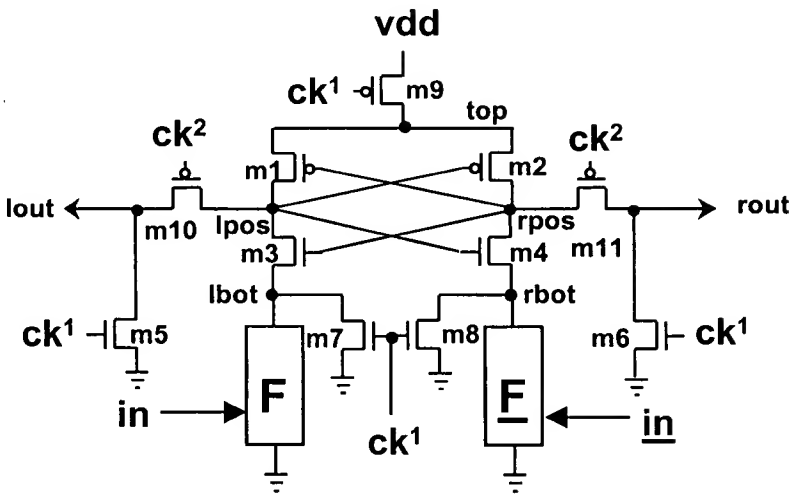

vdd
ck1
m9
top
ck2
ck2
m1
m2
lout
rout
m10
lpos
rpos
m11
m3
m4
lbot
rbot
ck1
m5
m7
m8
m6
ck1
F
F
in
ck1
in

Fig 10B

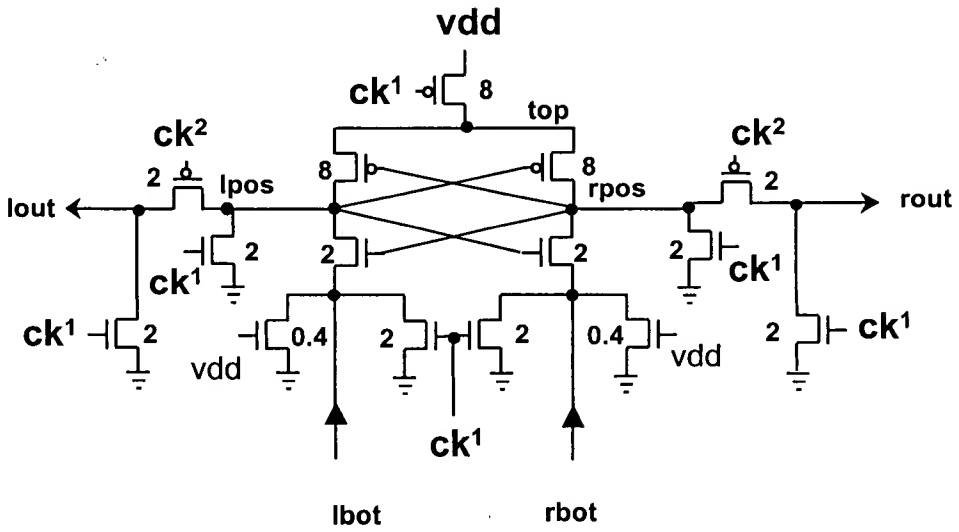

vdd
ck1
8
top
ck2
2
lpos
8
8
rpos
ck2
2
lout
rout
2
2
2
2
ck1
ck1
ck1
2
0.4
2
2
0.4
2
ck1
vdd
vdd
ck1
lbot
rbot

Fig 10C

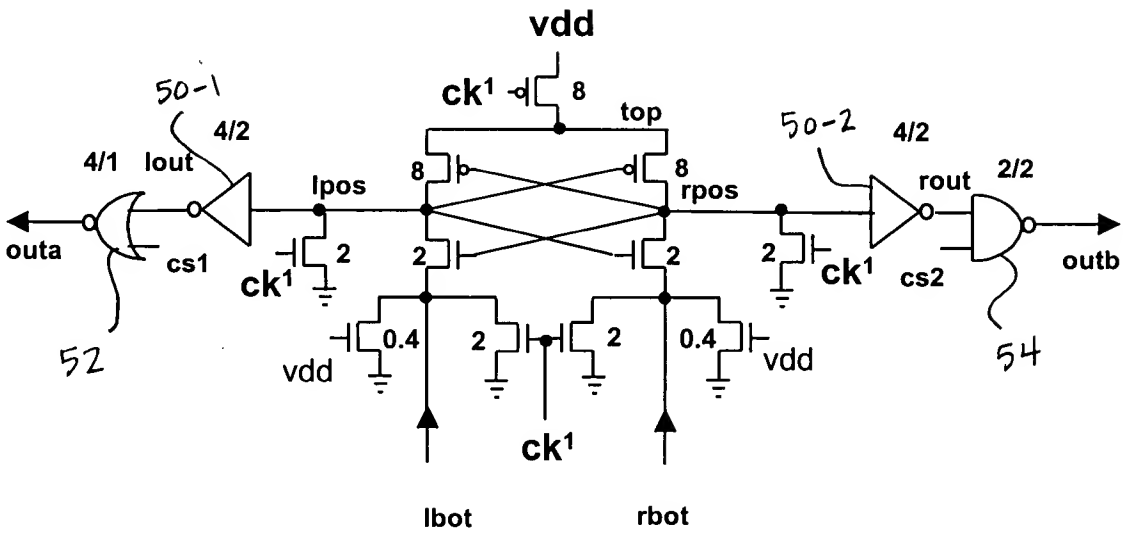

vdd
ck1
8
top
50-1
4/2
4/1
lout
lpos
8
8
rpos
50-2
4/2
rout
2/2
outa
cs1
2
2
2
2
ck1
ck1
cs2
outb
52
0.4
2
2
0.4
vdd
vdd
54
ck1
lbot
rbot

Fig 10D

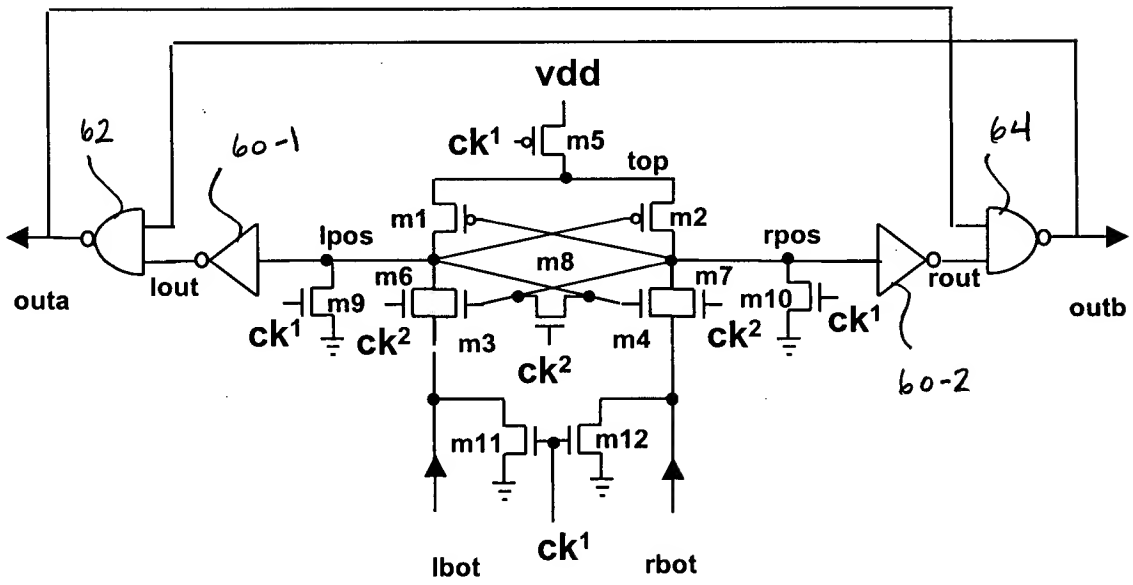


Fig 11

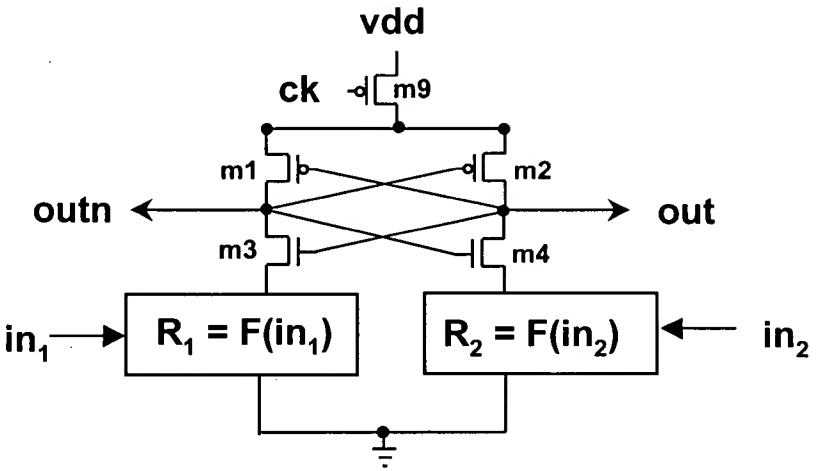


Fig 12

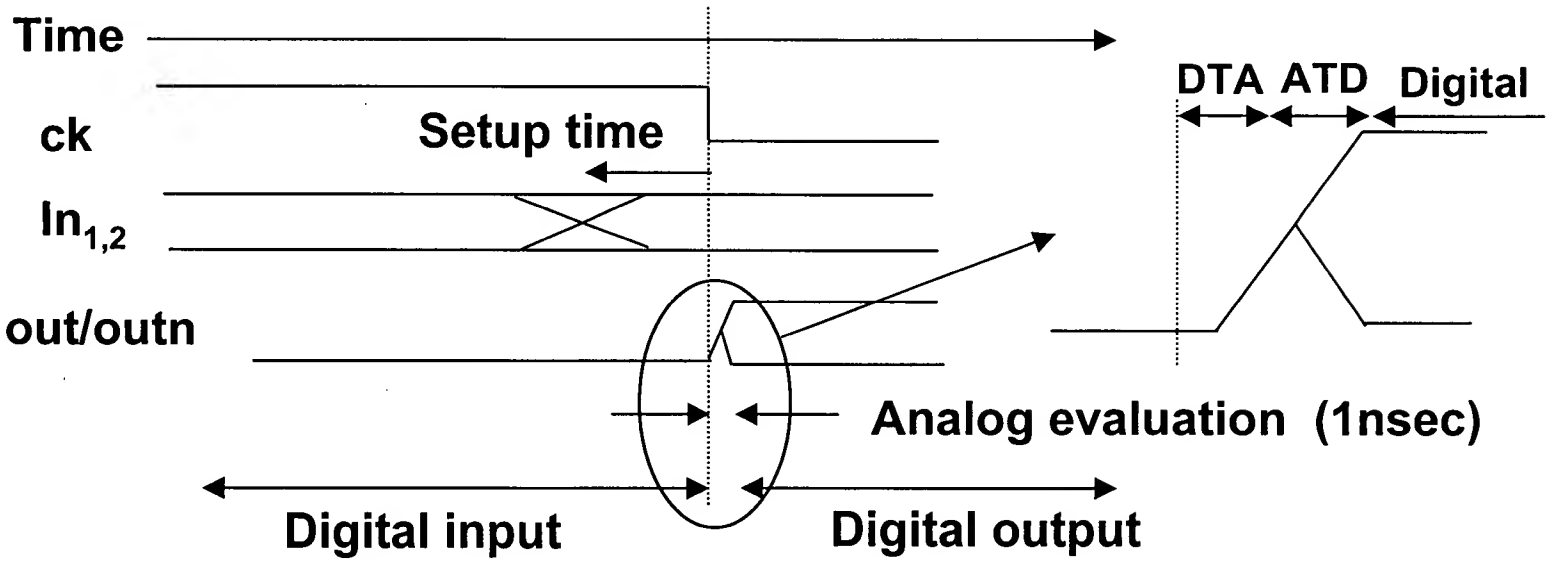

Time
ck
Setup time
In1,2
out/outn
DTA ATD Digital
Analog evaluation (1nsec)
Digital input
Digital output

Fig 13

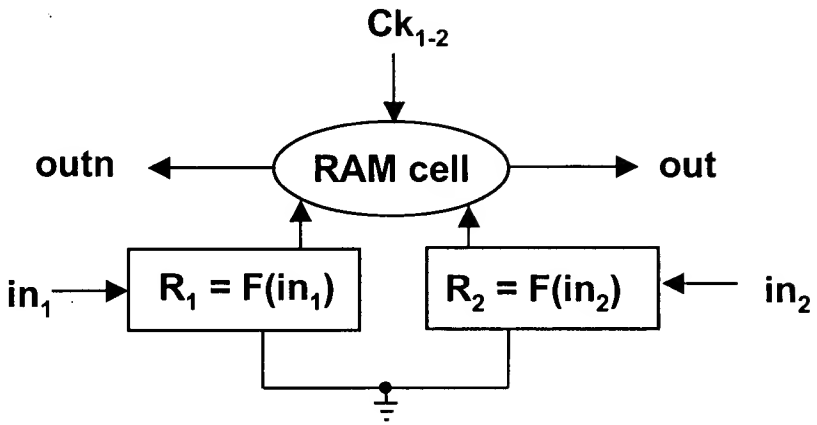


Fig 14

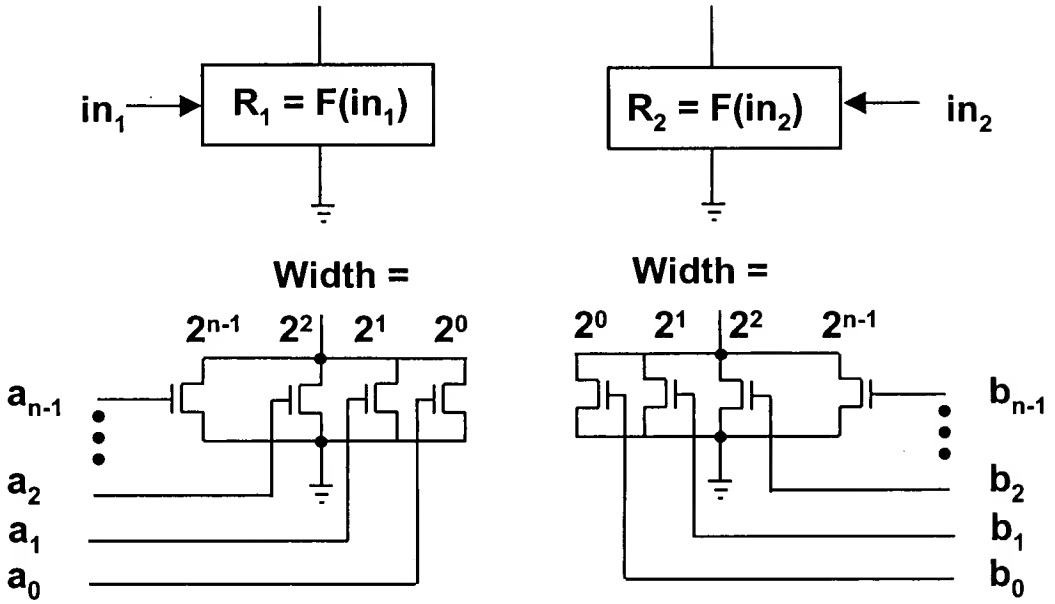

in1
R1 = F(in1)
R2 = F(in2)
in2
Width =
Width =
2n-1 22 21 20
20 21 22 2n-1
an-1
a2
a1
a0
bn-1
b2
b1
b0

Fig 15

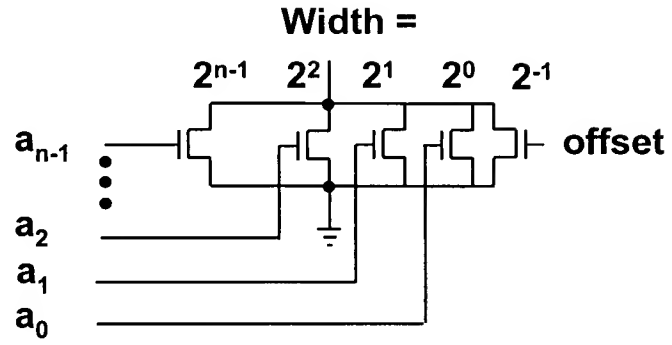


Fig 16

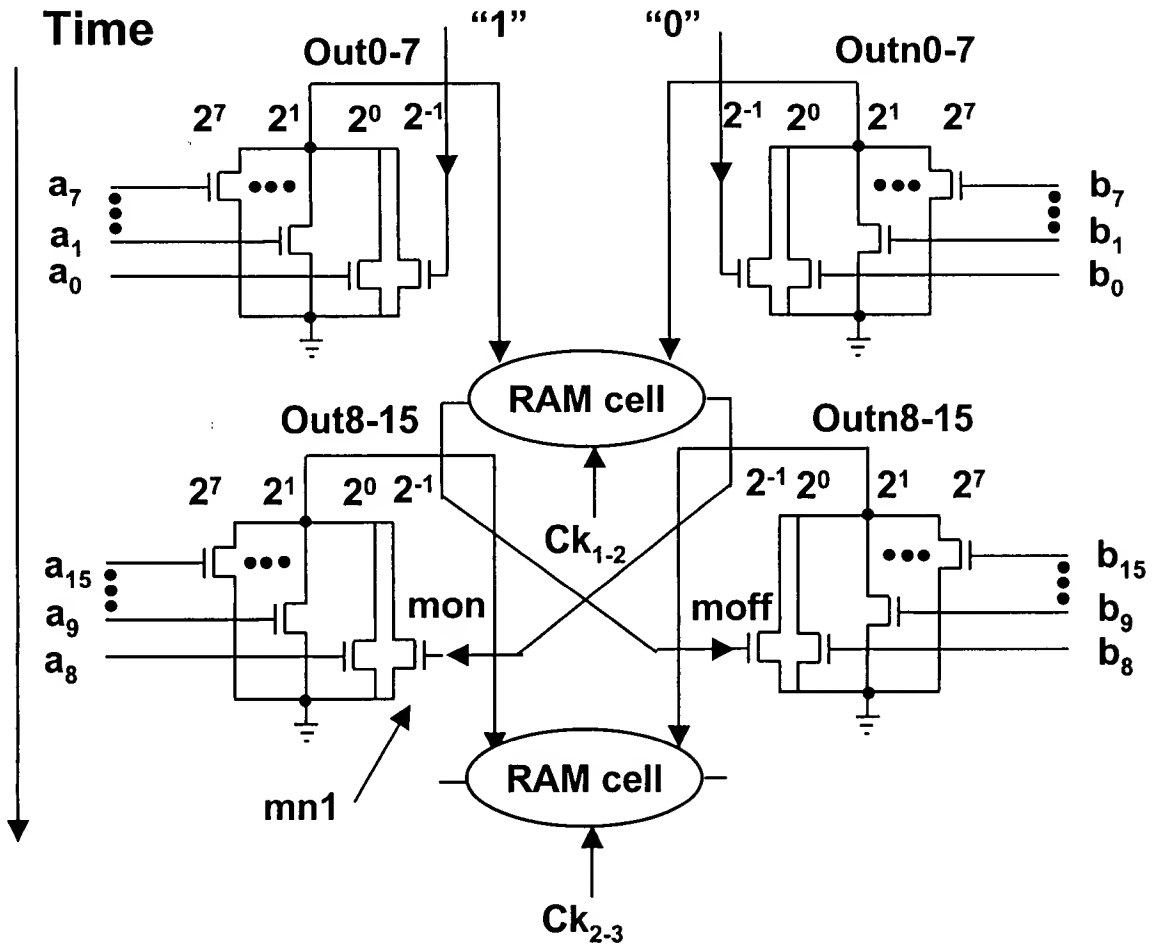
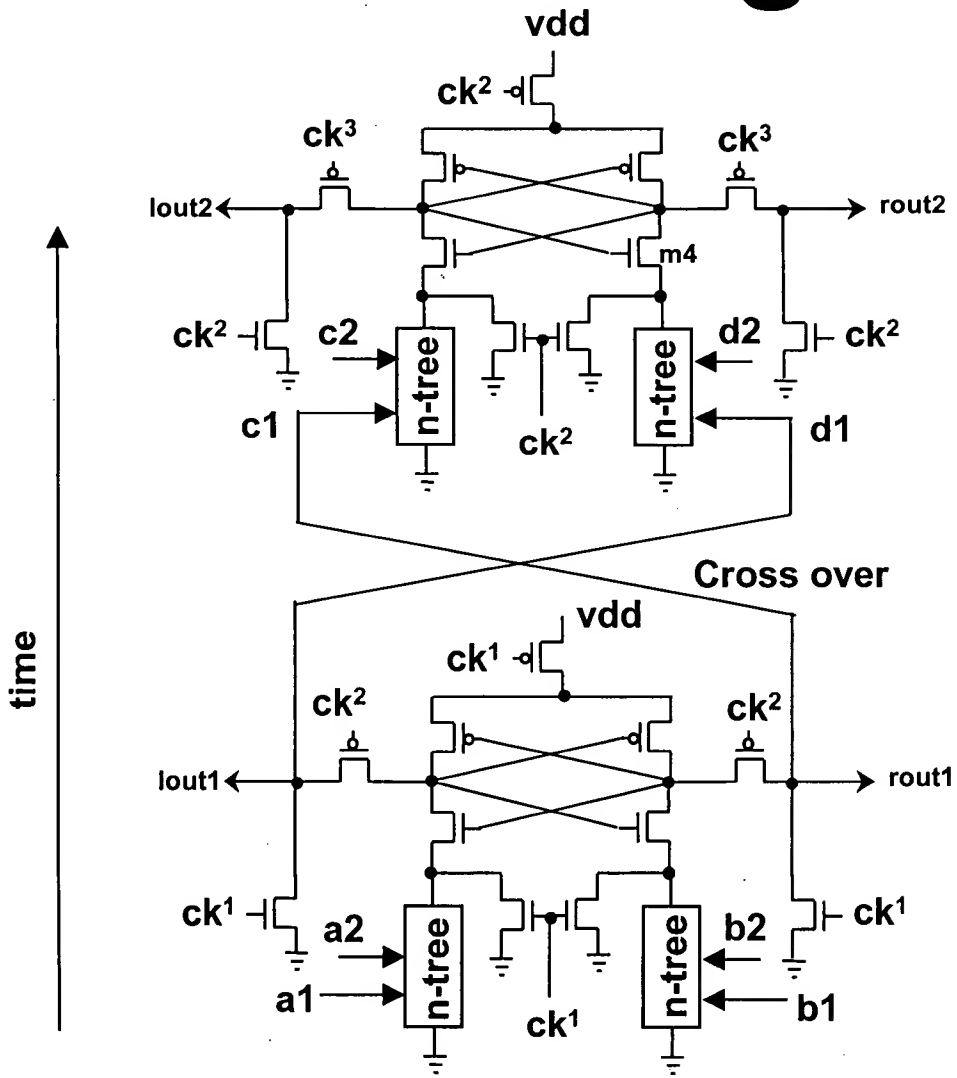


Fig 17


vdd
ck2
ck3
ck3
lout2
rout2
m4
ck2
c2
n-tree
d2
ck2
c1
ck2
n-tree
d1
Cross over
vdd
ck1
ck2
ck2
lout1
rout1
ck1
a2
n-tree
b2
ck1
a1
ck1
n-tree
b1
time

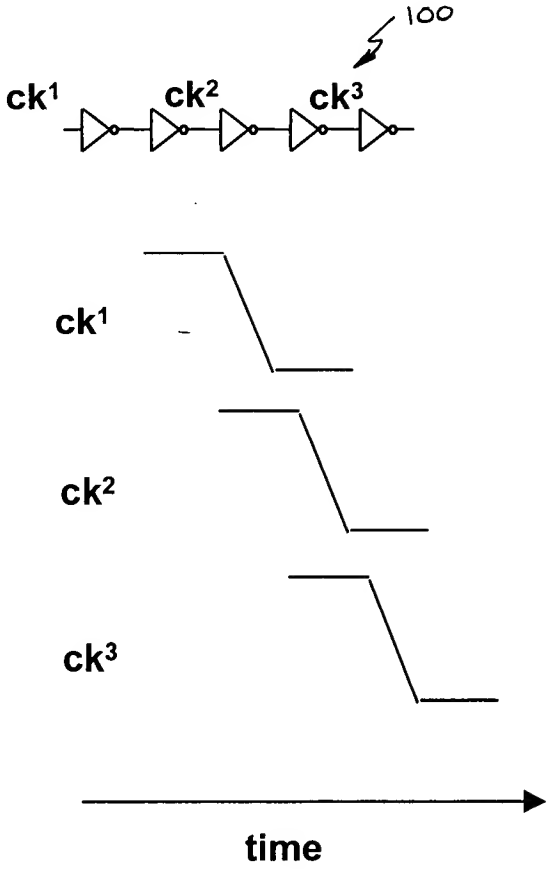
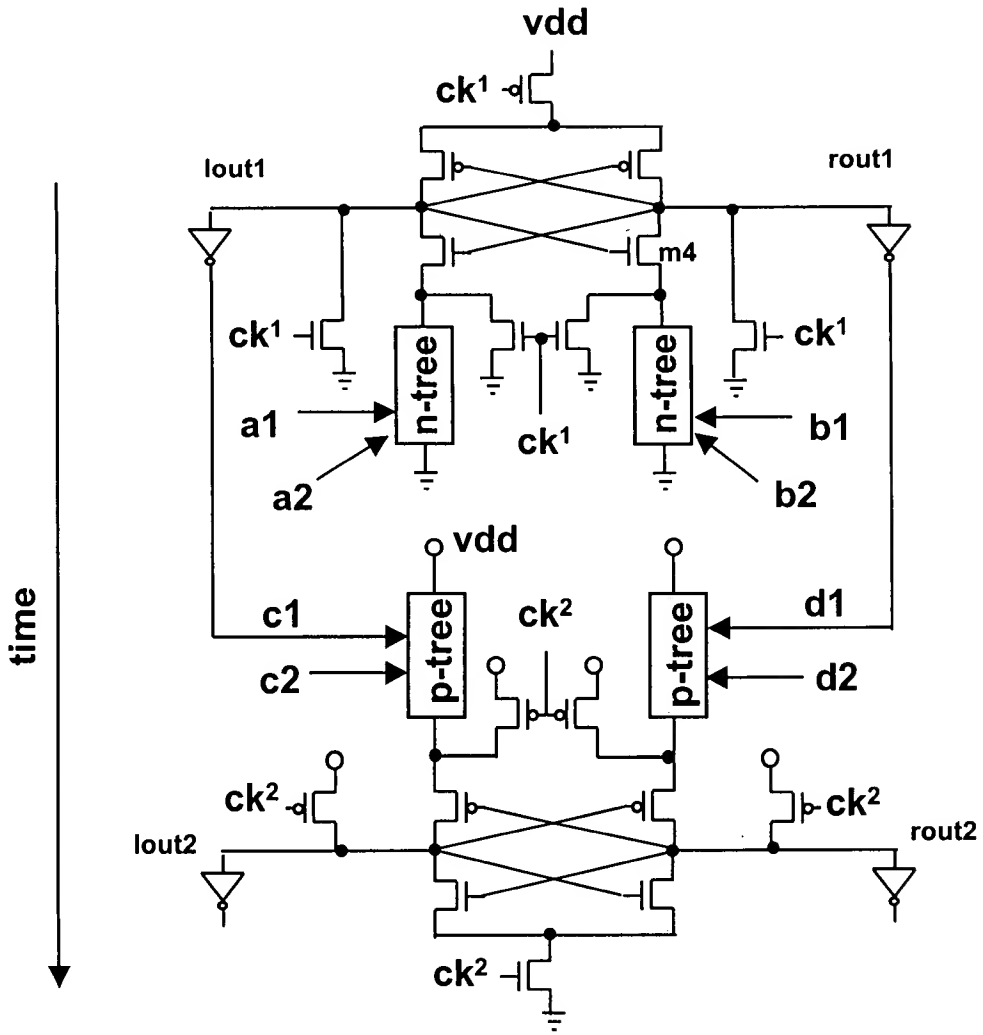

100
ck1
ck2
ck3
ck1
ck2
ck3
time

Fig 18


vdd
ck1
lout1
rout1
m4
ck1
ck1
n-tree
n-tree
ck1
a1
a2
b1
b2
vdd
c1
c2
ck2
p-tree
p-tree
d1
d2
ck2
ck2
lout2
rout2
ck2
time

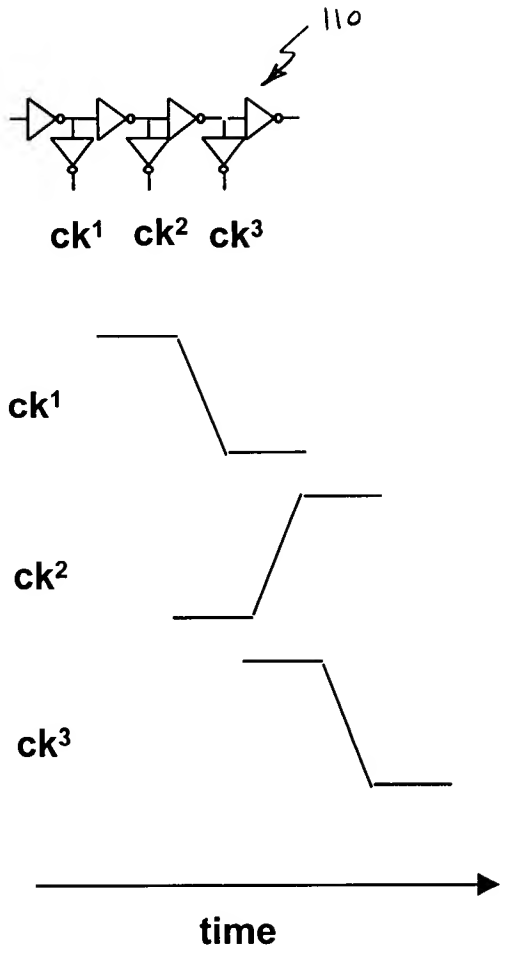

110
ck1 ck2 ck3
ck1
ck2
ck3
time

Fig 19

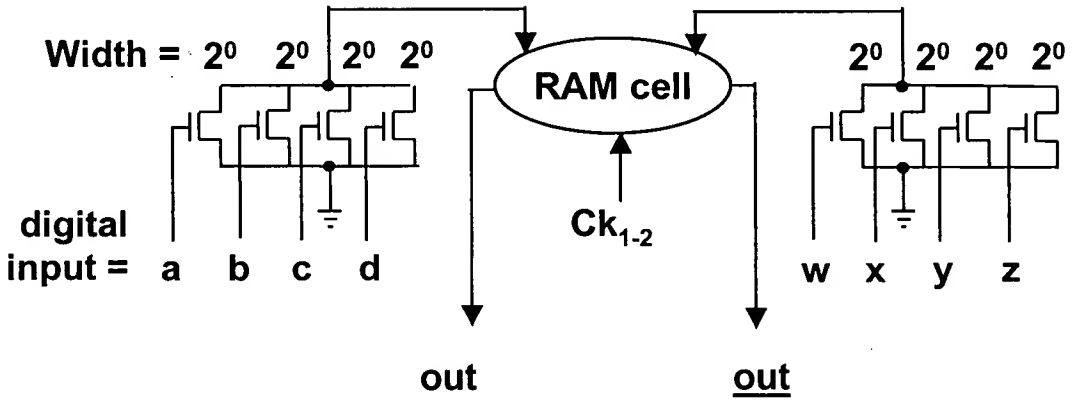

Width = 2⁰ 2⁰ 2⁰ 2⁰
RAM cell
2⁰ 2⁰ 2⁰ 2⁰
digital
input = a b c d
Ck1-2
w x y z
out
out

Fig 20

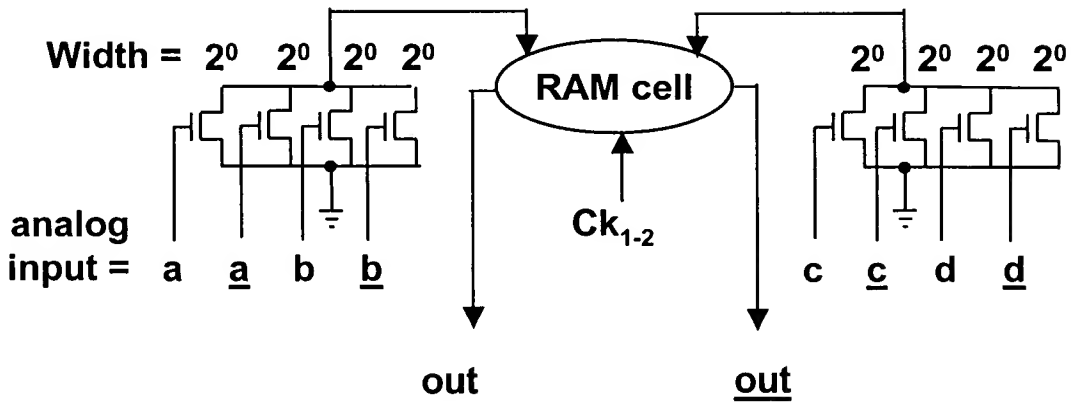

Width = 2⁰ 2⁰ 2⁰ 2⁰
RAM cell
2⁰ 2⁰ 2⁰ 2⁰
analog input = a a b b
Ck1-2
c c d d
out
out

Fig 21

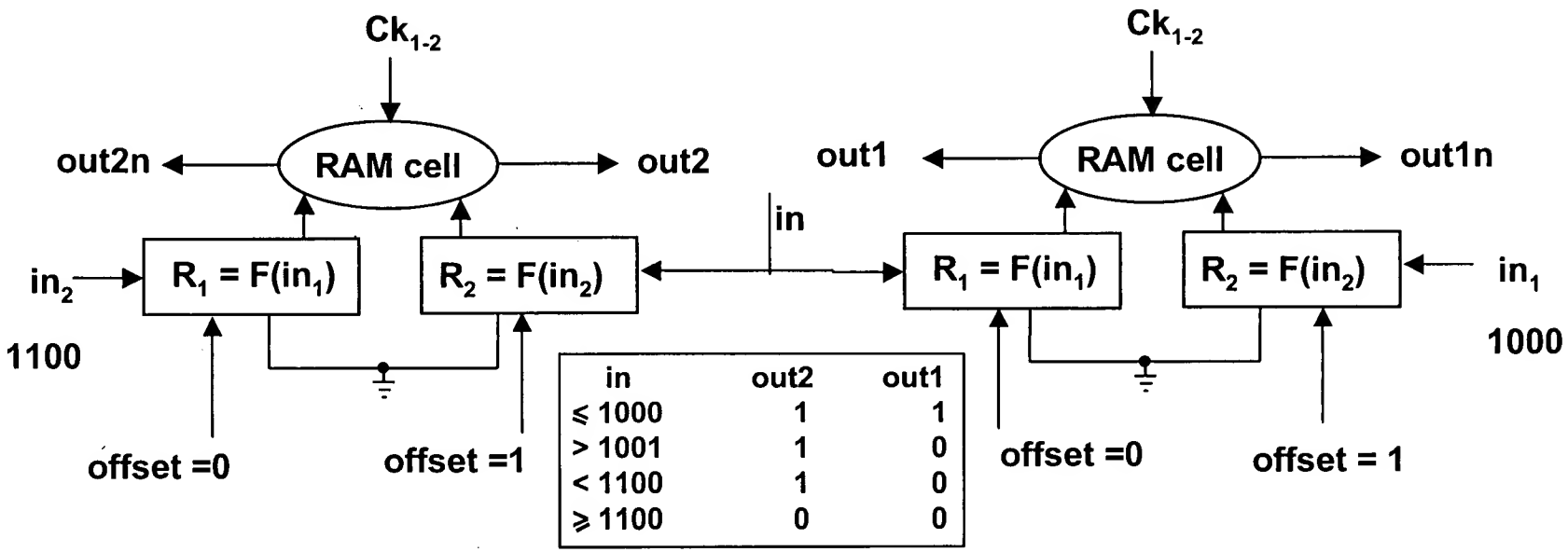


Fig 22

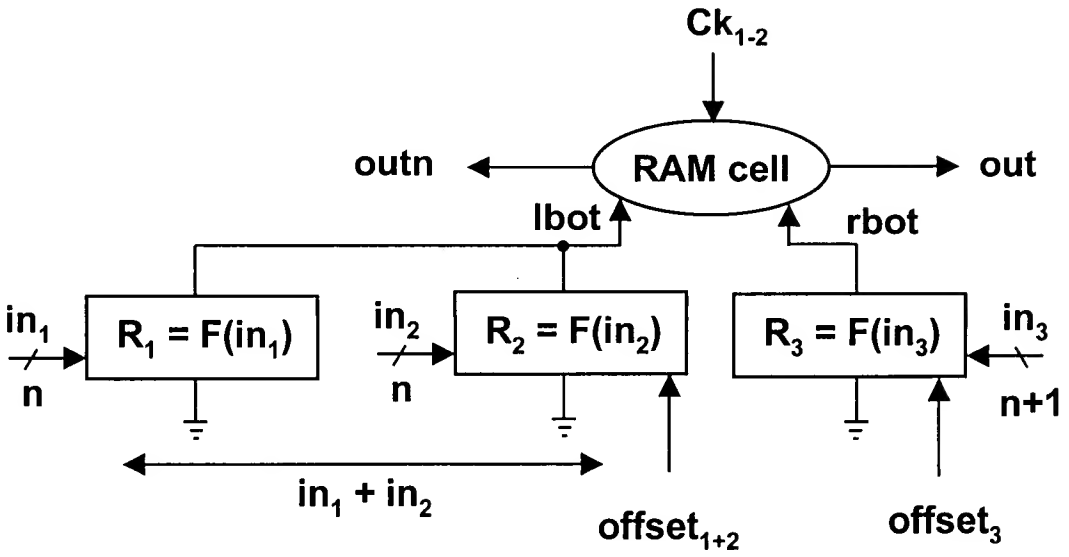

Ck1-2
outn
RAM cell
out
lbot
rbot
in1
n
R1 = F(in1)
in2
n
R2 = F(in2)
R3 = F(in3)
in3
n+1
in1 + in2
offset1+2
offset3

Fig 23

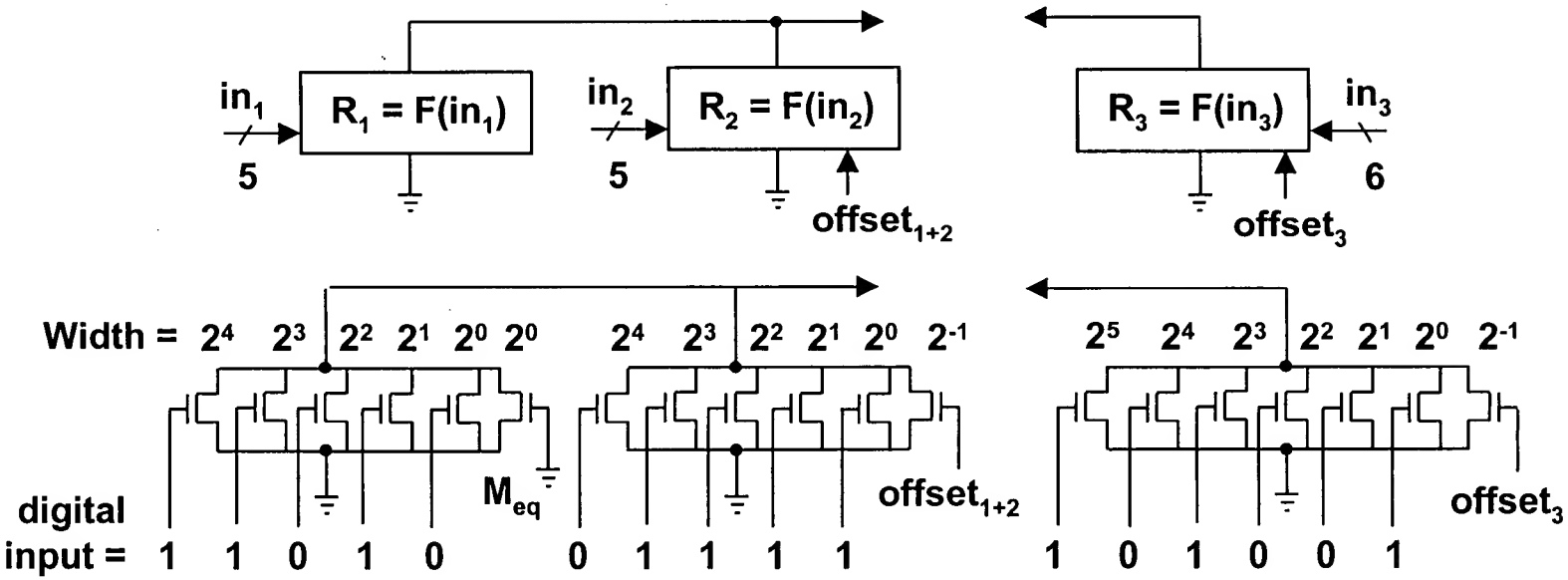

in1
5
R1 = F(in1)
in2
5
R2 = F(in2)
offset1+2
R3 = F(in3)
in3
6
offset3
Width =
2^4 2^3 2^2 2^1 2^0 2^0
2^4 2^3 2^2 2^1 2^0 2^-1
2^5 2^4 2^3 2^2 2^1 2^0 2^-1
Meq
offset1+2
offset3
digital
input =
1 1 0 1 0
0 1 1 1 1
1 0 1 0 0 1

Fig 24A

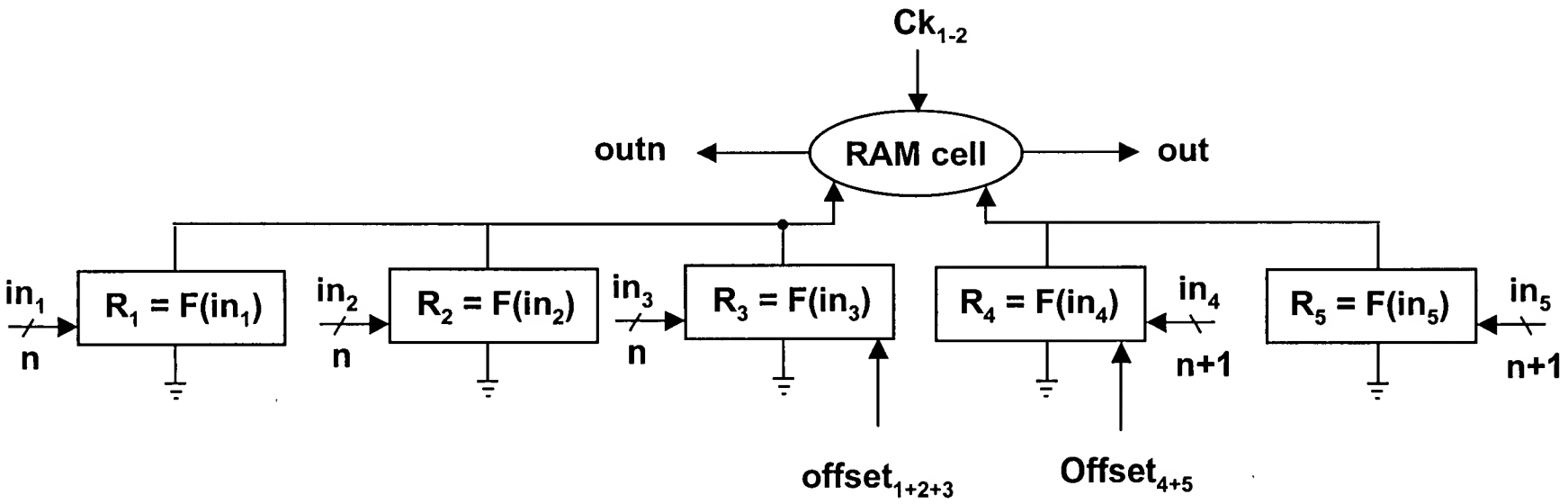

Ck1-2
outn
RAM cell
out
in1
n
R1 = F(in1)
in2
n
R2 = F(in2)
in3
n
R3 = F(in3)
R4 = F(in4)
in4
n+1
R5 = F(in5)
in5
n+1
offset1+2+3
Offset4+5

Fig 24B

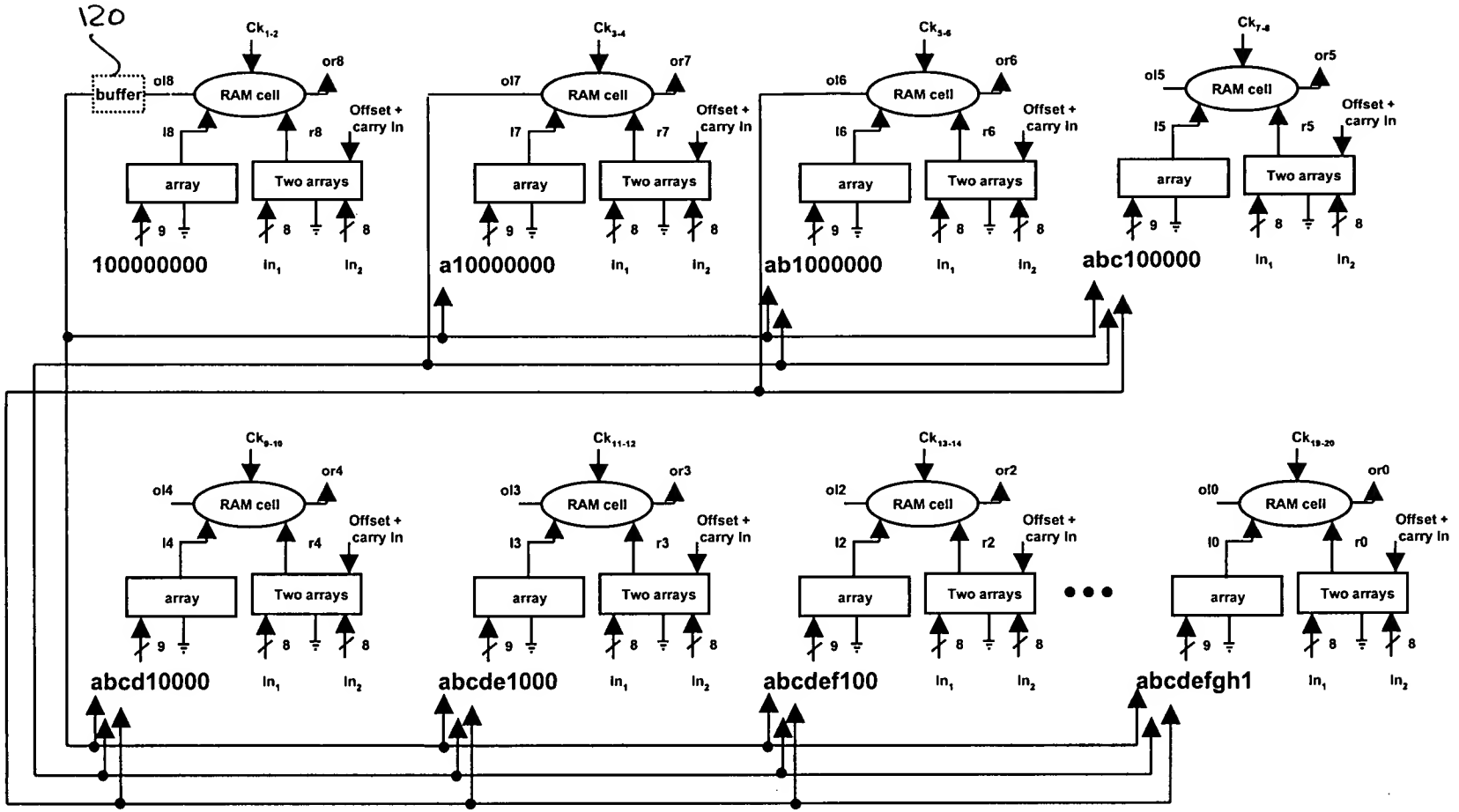

120
buffer
RAM cell
Offset + carry In
array
Two arrays
100000000
a10000000
ab1000000
abc100000
abcd10000
abcde1000
abcdef100
abcdefgh1

Fig 25

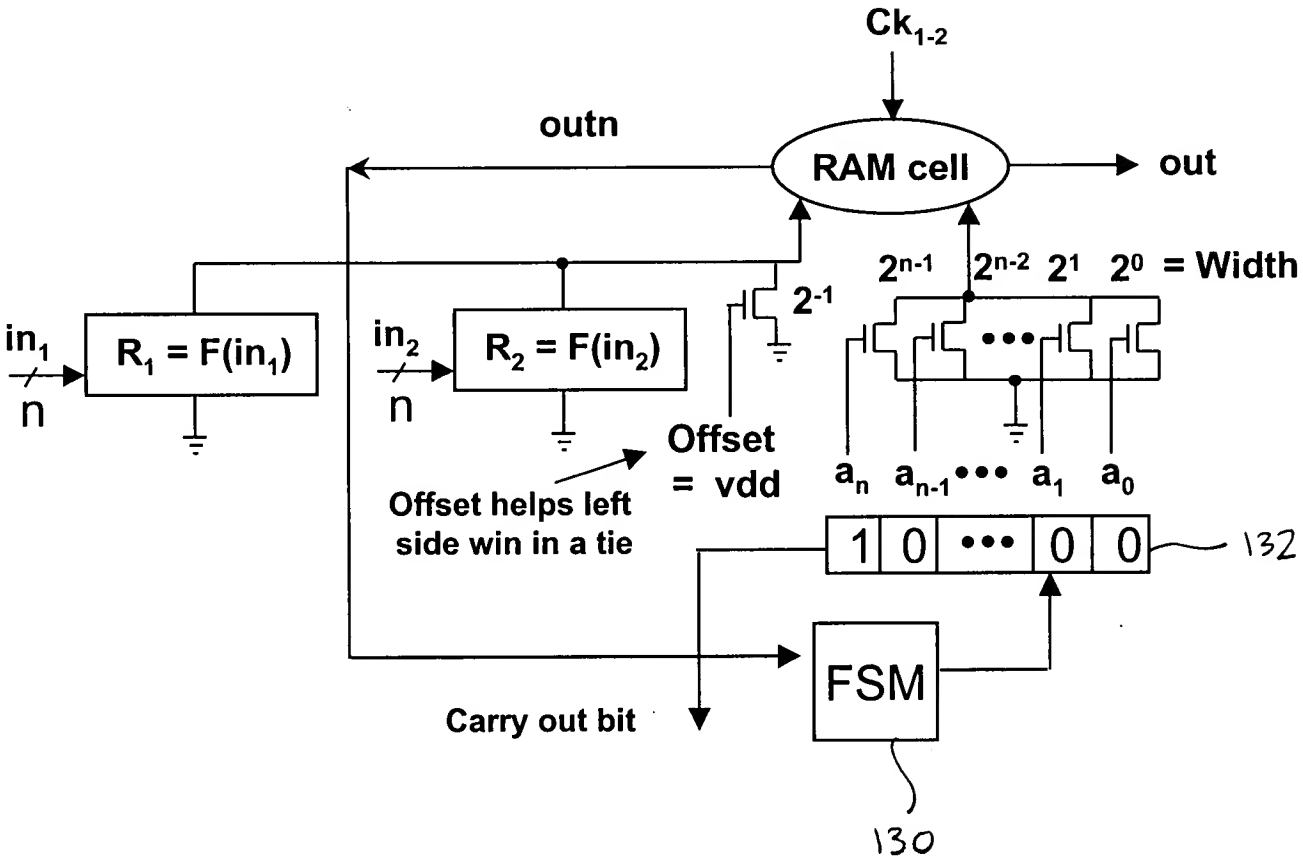

Ck1-2
outn
RAM cell
out
2n-1 2n-2 21 20 = Width
2-1
in1
n
R1 = F(in1)
in2
n
R2 = F(in2)
Offset
= vdd
Offset helps left
side win in a tie
an an-1 ••• a1 a0
1 0 ••• 0 0
132
FSM
Carry out bit
130

Fig 26

```
/* initialize */
n=8 ;
a[n] = 1;
for (k=1; k< n; k++)
   a[k]=0;

for (i=0; i<n; ++i)   /* i = clock tick */
  if (out = 0)
     a[n-i-1] =1;
  else {
     a[n-i] = 0;
        if ( i < n - 1)
          a[n-i-1] = 1;
  }
/* addition result is in array a[n] */
```

Fig 27

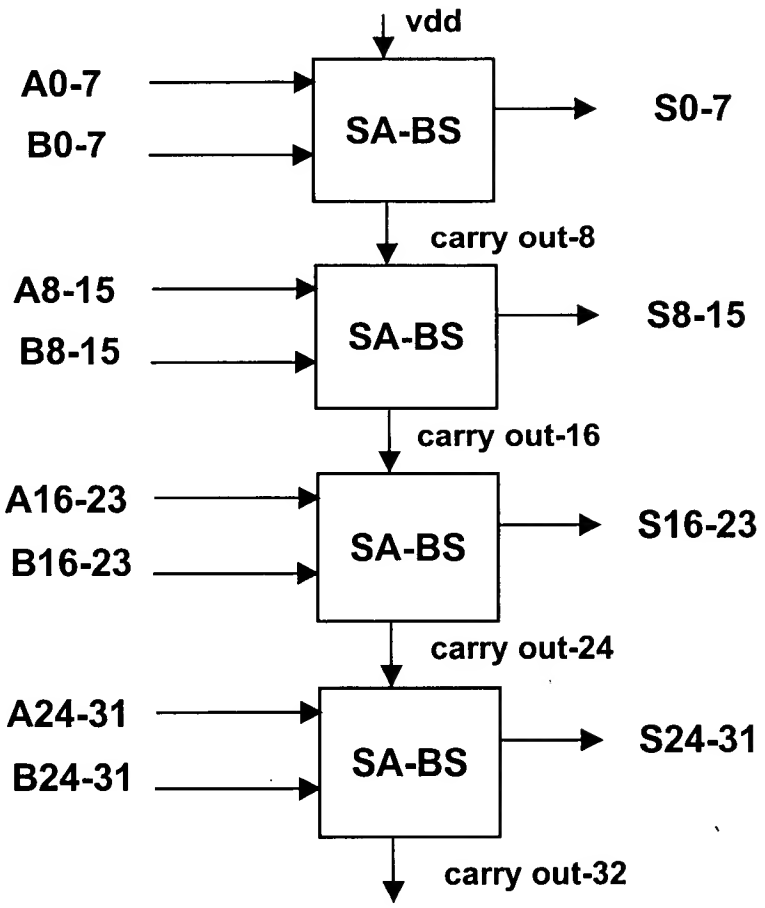


Fig 28

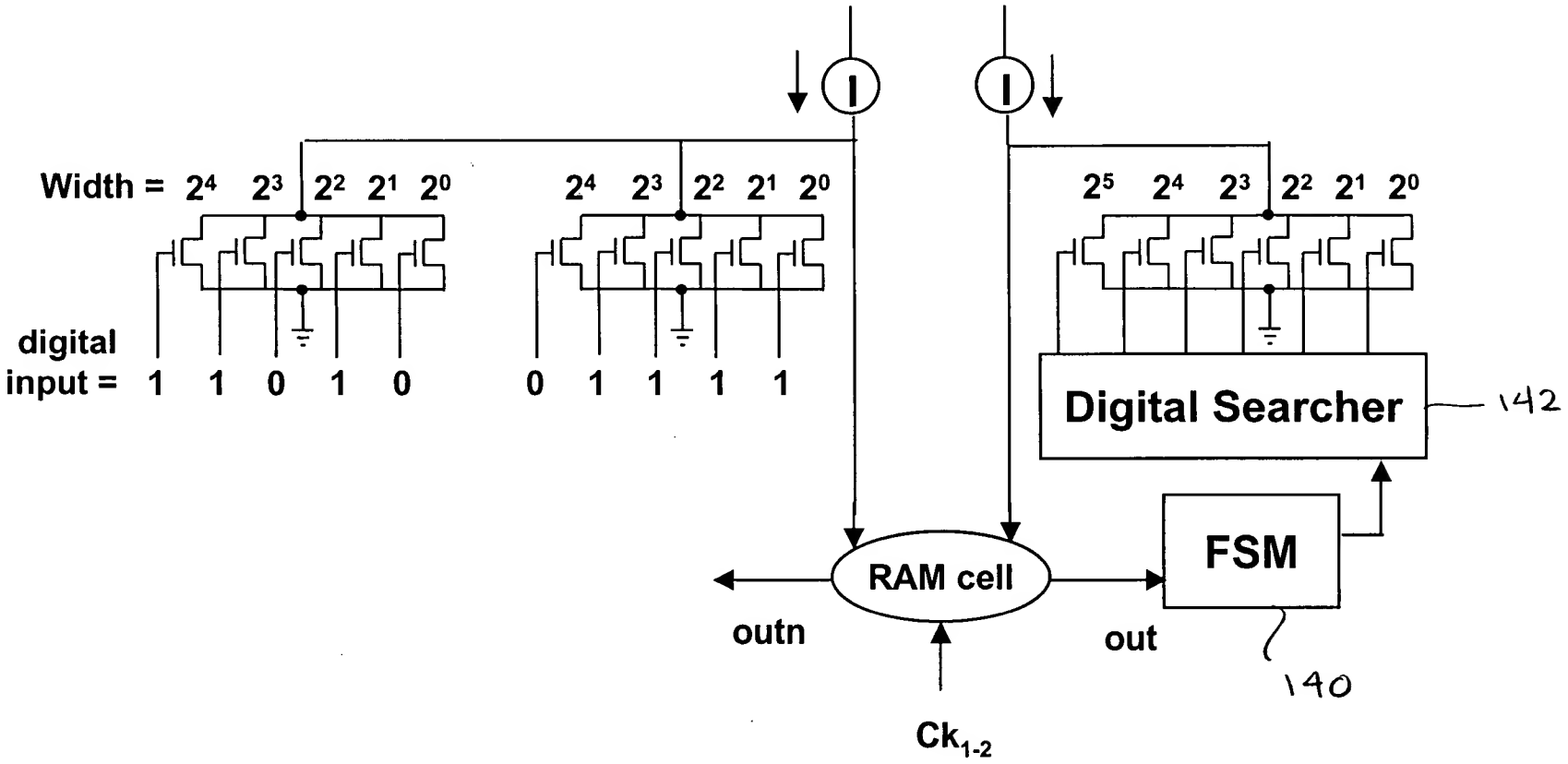
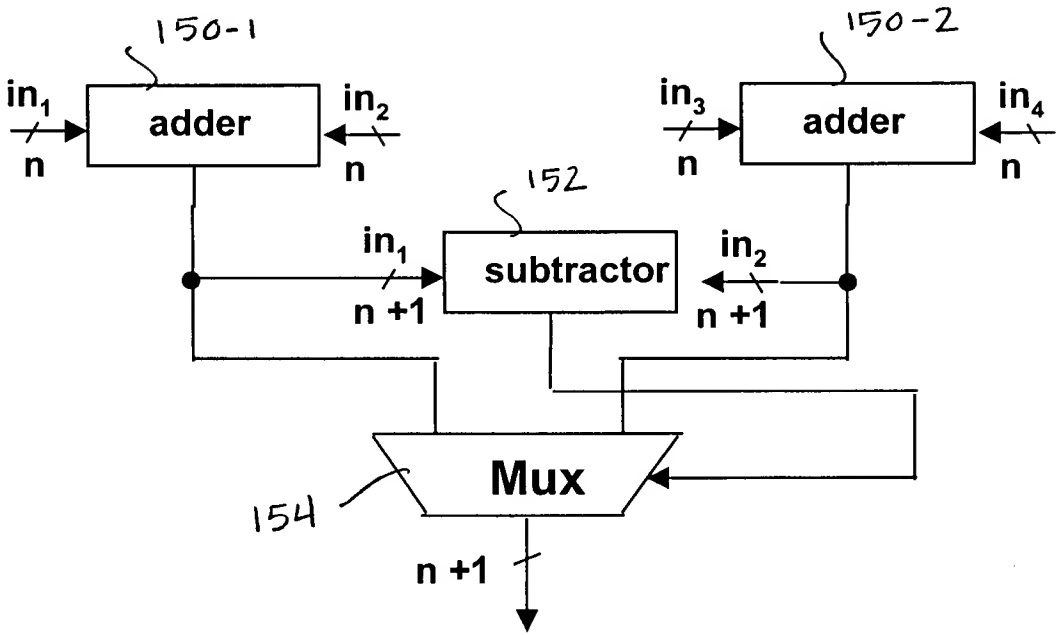

Width =
2^4 2^3 2^2 2^1 2^0
2^4 2^3 2^2 2^1 2^0
2^5 2^4 2^3 2^2 2^1 2^0
digital input =
1 1 0 1 0
0 1 1 1 1
I
I
Digital Searcher
142
FSM
RAM cell
outn
out
140
Ck_{1-2}

Fig 29



Prior Art

Fig 30

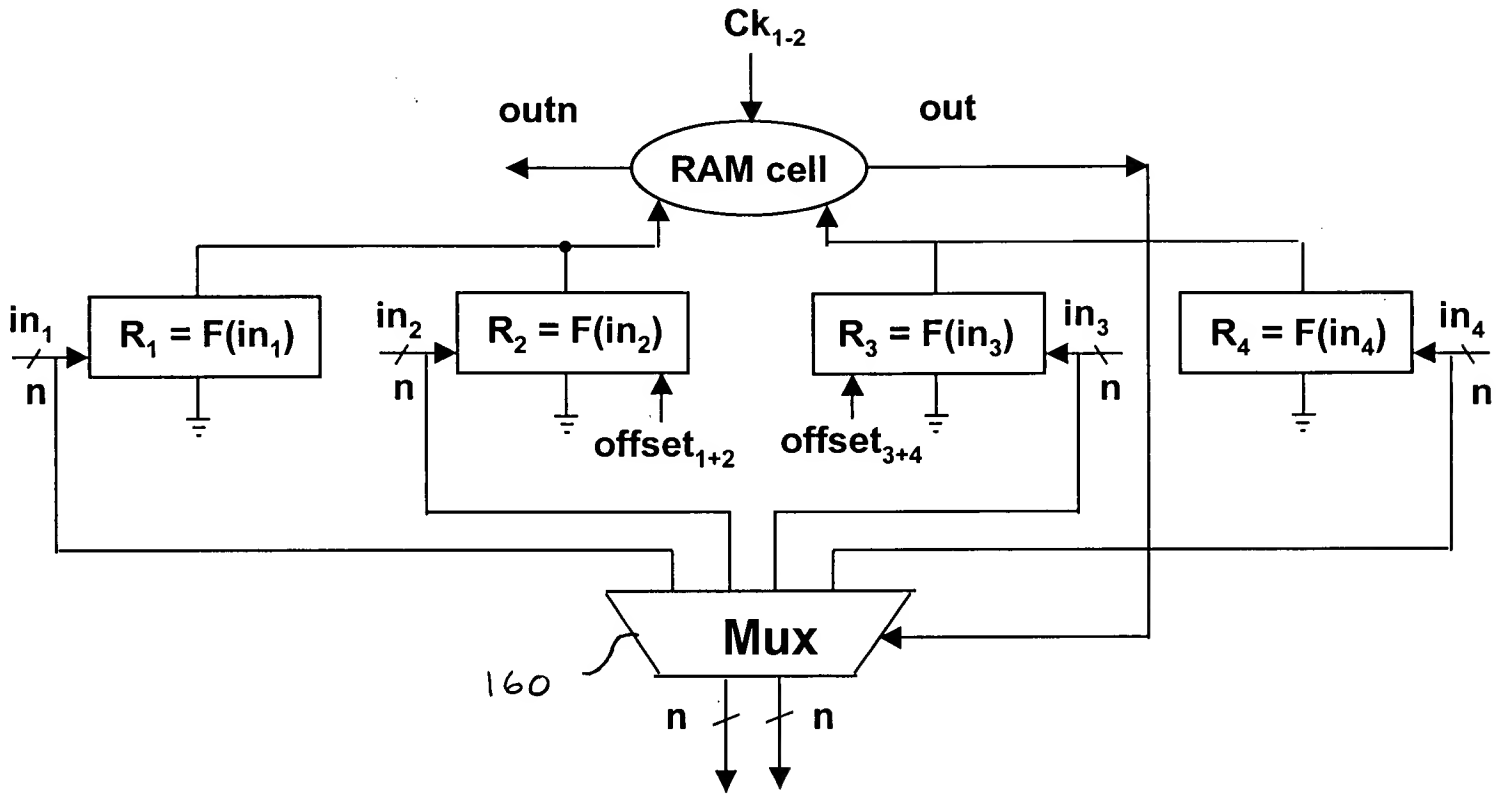

Ck1-2
outn
out
RAM cell
in1
n
R1 = F(in1)
in2
n
R2 = F(in2)
offset1+2
R3 = F(in3)
offset3+4
in3
n
R4 = F(in4)
in4
n
Mux
160
n
n

Fig 31

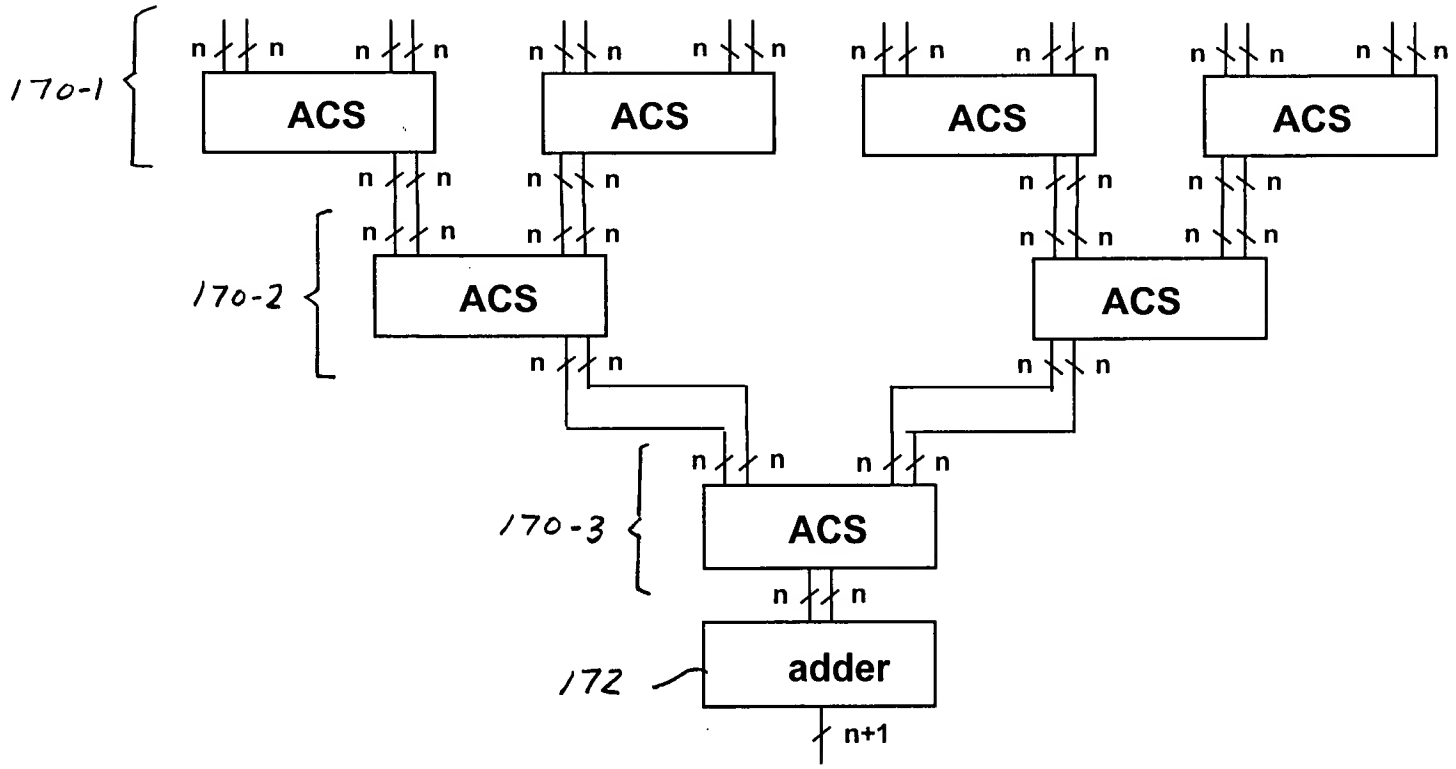


Fig 32

	POWER (μW)		TRANSISTOR COUNT		DELAY (pSEC)	
	Add Compare	Select	Add Compare	Select	Add Compare	Select
Conventional	2000	200	5215	58	1120	200
SeeSaw	100	300	55	108	516	173
Gain Factor	X20	X0.66	X95	X0.54	X2.17	X1.15

Fig 33

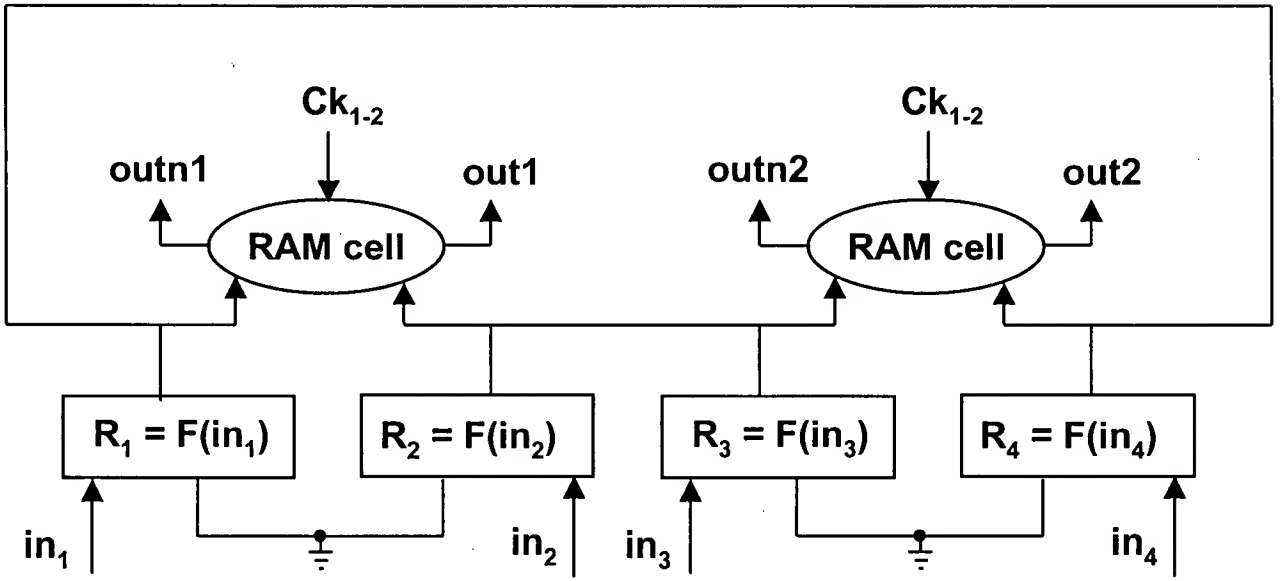

Ck1-2
outn1
out1
RAM cell
Ck1-2
outn2
out2
RAM cell
R1 = F(in1)
R2 = F(in2)
R3 = F(in3)
R4 = F(in4)
in1
in2
in3
in4

Fig 34

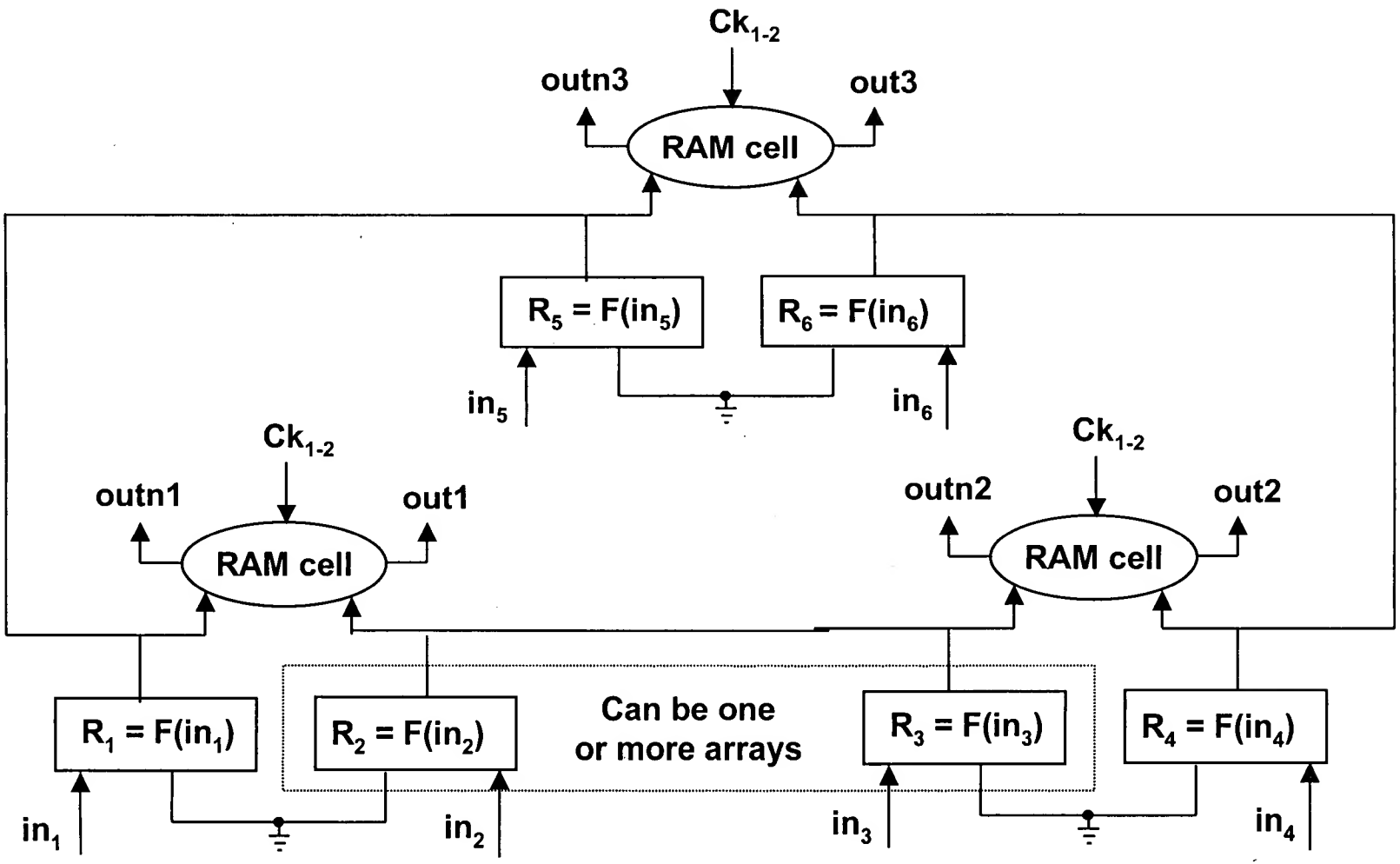

Ck1-2
outn3
out3
RAM cell
R5 = F(in5)
R6 = F(in6)
in5
in6
Ck1-2
outn1
out1
RAM cell
Ck1-2
outn2
out2
RAM cell
R1 = F(in1)
R2 = F(in2)
Can be one
or more arrays
R3 = F(in3)
R4 = F(in4)
in1
in2
in3
in4

Fig 35

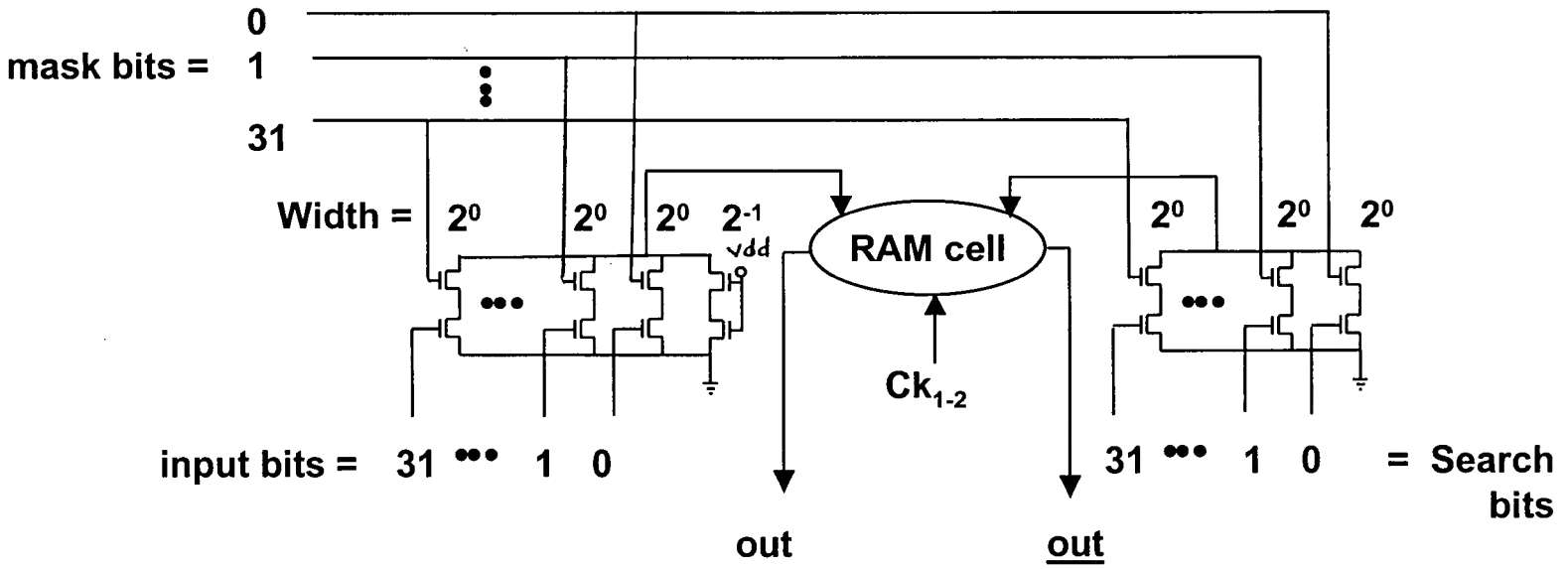

0
mask bits = 1
31
Width = 2^0 2^0 2^0 2^-1
Vdd
RAM cell
2^0 2^0 2^0
Ck1-2
input bits = 31 ••• 1 0
31 ••• 1 0 = Search bits
out
out

Fig 36